

WHAT IS CLAIMED IS:

Interrupt driven value sampling

1 A method for value sampling for monitoring the performance of a program  
2 being executed on a computer system, comprising the steps of:  
3 executing the program on a computer system, the program having object code  
4 instructions;  
5 at intervals interrupting execution of the program, including delivering a first  
6 interrupt; and  
7 in response to at least a subset of the first interrupts, storing at least one data  
8 value of interest in a first database, the at least one data value of interest being  
9 associated with a particular object code instruction of the program, the particular object  
10 code instruction being executed by the computer, such that the program remains  
11 unmodified.

1 2. The method of claim 1 wherein the intervals are random intervals.

1 3. The method of claim 1 wherein the intervals have a constant period.

1 4. The method of claim 1 further comprising the step of:  
2 specifying an object code instruction of interest, and wherein the at least one  
3 data value of interest is associated with the specified object code instruction of interest.

1 5. The method of claim 1 wherein the data value of interest is an operand.

1 6. The method of claim 1 wherein the data value of interest is a result of the  
2 execution of the instruction.

1 7. The method of claim 1 wherein the particular object code instruction is  
2 associated with a memory address, the memory address being stored in a program

On the 11th of June 1862, I left with four loads of hay and some corn at

On the 11th of June 1862, I left with four loads of hay and some corn at

On the 11th of June 1862, I left with four loads of hay and some corn at

On the 11th of June 1862, I left with four loads of hay and some corn at

On the 11th of June 1862, I left with four loads of hay and some corn at

On the 11th of June 1862, I left with four loads of hay and some corn at

On the 11th of June 1862, I left with four loads of hay and some corn at

On the 11th of June 1862, I left with four loads of hay and some corn at

1 branch instruction, a jump instruction, a call subroutine instruction, or a subroutine  
2 return instruction, and the destination is identified by an associated destination address.

1 15. The method of claim 13 wherein the at least one instruction is a  
2 conditional branch instruction, and the destination is identified by a bit.

1 16. The method of claim 1 wherein said step of storing includes the steps of:  
2 identifying at least one issue block of instructions;  
3 interpreting the instructions of the at least one issue block; and  
4 storing at least one data value of interest associated with at least one interpreted  
5 instruction.

1 17. The method of claim 16 wherein said step of storing stores the at least  
2 one data value of interest after interpreting each instruction of said at least one issue  
3 block.

1 18. The method of claim 16 wherein said step of interpreting emulates a  
2 machine language instruction set of the computer system.

1 19. The method of claim 16 wherein the interpreter updates a state of the  
2 interrupted program as though each interpreted instruction had been directly executed  
3 by the computer system.

1 20. The method of claim 16 wherein the interpreter interprets both kernel and  
2 user code.

1 21. The method of claim 16 wherein the interpreter does not execute  
2 particular instructions.

Sub A4 7

1 22. The method of claim 1 wherein the particular object code instruction is an  
2 interrupted instruction, the particular object code instruction being associated with a  
3 memory address, and said step of storing includes the steps of:  
4 storing the memory address and the interrupted instruction;  
5 configuring a second interrupt to be delivered after a predetermined number of  
6 instructions are executed; and  
7 in response to the second interrupt,  
8 deactivating the second interrupt, and  
9 storing the memory address and the at least one data value of interest for  
10 the associated instruction in the first database.

1 23. The method of claim 22 wherein the predetermined number of  
2 instructions is equal to a number of instructions of an issue block.

1 24. The method of claim 1 wherein the particular object code instruction is an  
2 interrupted instruction, the particular object code instruction being associated with a  
3 memory address, and said step of storing includes the steps of:  
4 storing the memory address and a set of object code instructions including the  
5 interrupted instruction;  
6 configuring a second interrupt to be delivered after a predetermined number of  
7 instructions; and  
8 in response to the second interrupt,  
9 deactivating the second interrupt, and  
10 storing the memory address and the at least one data value of interest for  
11 the interrupted instruction in the first database.

1 25. The method of claim 24 further comprising the step of:  
2 analyzing the interrupted instruction to determine the at least one data value of  
3 interest to store in the first database.

[illegible][illegible]

**Wish**

U

1           32.    The method of claim 1 wherein the at least one data value of interest has  
2 a plurality of data values of interest, and said step of storing stores a tuple of the  
3 plurality of the data values of interest and a memory address in the first database.

1           33.    The method of claim 32 wherein one of the data values of interest is a  
2 particular data value of interest and the tuple includes a count that is associated with  
3 the particular data value of interest, and  
4 said step of storing further includes the steps of:  
5           identifying the particular data value of interest in the first database; and  
6           incrementing the count associated with the particular data value of interest.

1           34.    The method of claim 32 wherein the data values of interest stored in the  
2 tuple include a value stored in a register specified by the interrupted instruction.

1           35. The method of claim 32 wherein the data values of interest stored in the  
2 tuple include the value stored in a destination register specified by the instruction and  
3 at least one other value stored in another register.

1           36. The method of claim 1 wherein the particular object code instruction is part  
2 of an issue block of instructions, and said step of storing stores one or more data  
3 values of interest for one or more instructions of the issue block, the data values of  
4 interest including a value of a register specified by one of the instructions of the issue  
5 block, including a value stored in a return address register and a value stored in a  
6 memory location in a current stack frame.

1           37.    The method of claim 32, wherein one of the data values of interest is a  
2 particular data value of interest and the tuple is a particular tuple having particular  
3 values of interest, the tuple also having a count that is associated with all the particular  
4 data values of interest of the particular tuple, and  
5 said step of storing further includes the steps of:

**THE UNIVERSITY OF CHICAGO**

- 1
- 2
- 3
- 4
- 5

- 1
- 2
- 3
- 4
- 5
- 6
- 7

- 1
- 2
- 3

- 1
- 2
- 3

1  
2  
3  
4  
5

[illegible]

1            47.    The method of claim 32, wherein one of the data values of interest is a  
2    particular data value of interest and the tuple is a particular tuple having particular



Sub A47

values of interest, the tuple also storing a functional value, wherein the first database stores a set of tuples that are associated with different particular data values of interest associated with the particular object code instruction, for the particular object code instruction, updating the functional value in the particular tuple based on the at least one data value of interest associated with the particular object code instruction and at least one data value of interest of each tuple in the set of tuples.

48. The method of claim 16 further including the steps of:  
receiving a downloaded script; and

wherein said step of storing includes the steps of:

after interpreting one of the instructions, executing the downloaded script to determine and store the at least one data value of interest.

49. The method of 16 further including the steps of:  
receiving an interpretable program having interpretable instructions;  
receiving a downloaded script that causes a user-mode trap to be sent to the interpretable program;  
executing the interpretable instructions with a virtual machine; and  
wherein said step of interpreting interprets said executing interpretable instructions, wherein said interrupted instruction is derived from said interpretable instructions, and

said step of storing includes the step of executing the downloaded script to cause a user-mode trap to be sent to said interpretable program whereby virtual machine specific context information may be stored.

50. The method of 16 further including the steps of:  
receiving a JAVA program having bytecode instructions;  
receiving a downloaded script that causes a user-mode trap to be sent to the JAVA program;  
executing the bytecode instructions with a JAVA virtual machine; and

[illegible]

1            55.    The method of claim 1 wherein said step of storing includes the steps of:  
2            identifying access control identifiers associated with the program, particular ones  
3            of the access control identifiers also being associated a particular user; and  
4            storing the at least one data value such that such that only the particular user  
5            associated with the access control identifier can access the at least one data value.

Sub A4 7

1 56. The method of claim 55 wherein the access control identifier includes at  
2 least one of a process identifier, a user identifier and a group identifier.

1 57. The method of claim 1 wherein said step of storing includes the steps of:  
2 identifying a group identifier associated with the program; and  
3 storing the at least one data value of interest in a user space associated with the  
4 group identifier such that only a user associated with the group identifier can access  
5 the at least one data value.

1 58. The method of claim 1 further comprising the steps of:  
2 storing the at least one data value of interest of the first database in a hotlist of  
3 most frequently occurring data values, the hotlist storing a predetermined number of  
4 data values and a count associated with each data value.

1 59. The method of claim 58 further comprising the step of encoding the  
2 hotlist.

1 60. The method of claim 59 wherein said step of encoding encodes the hotlist  
2 by sorting at least a subset of the data values of the hotlist by the count.

1 61. The method of claim 59 wherein said step of encoding encodes the hotlist  
2 by reordering at least a subset of the data values of the hotlist such that the data  
3 values in the the at least one subset are stored in contiguous memory locations.

1 62. The method of claim 58 wherein said step of storing the at least one data  
2 value of interest of the first database stores the at least one data value of interest using  
3 a randomized technique.

1 63. The method of claim of claim 62 wherein said step of storing the at least  
2 one data value of interest dynamically adapts between a first randomized technique  
3 and a second randomized technique.

Sub A7

1 64. The method of claim 58 wherein said step of storing the at least one data  
2 value of interest of the first database in a hotlist of most frequently occurring data  
3 values uses a concise samples technique.

1 65. The method of claim 58 wherein said step of storing the at least one data  
2 value of interest of the first database in a hotlist of most frequently occurring data  
3 values uses a counting samples technique.

1 66. A computer system for value sampling a computer program having object  
2 code instructions while the object code instructions are executing without modifying the  
3 computer program, comprising:  
4 a processor for executing the object code instructions of the computer program;  
5 and  
6 a memory for storing instructions that:  
7 deliver interrupts at intervals during execution of the program, including  
8 delivering a first interrupt; and  
9 store at least one data value of interest in a first database in response to  
10 at least a subset of the first interrupts.

1 67. The computer system of claim 66 wherein said instructions that store  
2 further include instructions that:  
3 identify at least one issue block of instructions;  
4 interpret the instructions of the at least one issue block; and  
5 store at least one data value of interest associated with at least one interpreted  
6 instruction.

1 68. The computer system of claim 66, wherein said instructions that deliver  
2 interrupts interrupt a particular object code instruction as an interrupted instruction, the  
3 particular object code instruction being associated with a memory address; and  
4 wherein said instructions that store further include instructions that:

[illegible]

1        70.    The computer program product of claim 69 wherein the intervals are  
2    random intervals.

1           72.    The computer program product of claim 69 wherein the instructions that  
2    deliver interrupts interrupt a particular object code instruction as an interrupted

